CANreal

CANopen Tool for Testing and Monitoring CANopen Networks

Software Manual

esd electronic system design gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
www.esd-electronics.com • Fax: 0511/37 29 8-68
Phone: 0511/37 29 80 • International: +49-5 11-37 29 80

N O T E

The information in this document has been carefully checked and is believed to be entirely reliable. **esd** makes no warranty of any kind with regard to the material in this document, and assumes no responsibility for any errors that may appear in this document. **esd** reserves the right to make changes without notice to this, or any of its products, to improve reliability, performance or design.

esd assumes no responsibility for the use of any circuitry other than circuitry which is part of a product of **esd** gmbh.

esd does not convey to the purchaser of the product described herein any license under the patent rights of **esd** gmbh nor the rights of others.

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone: +49-511-372 98-0
Fax: +49-511-372 98-68
E-mail: info@esd-electronics.com
Internet: www.esd-electronics.com

USA / Canada:
esd electronics Inc.
525 Bernardston Road
Suite 1
Greenfield, MA 01301
USA

Phone: +1-800-732-8006
Fax: +1-800-732-8093
E-mail: us-sales@esd-electronics.com
Internet: www.esd-electronics.us

Document file:	I:\texte\Doku\MANUALS\PROGRAM\CAN\CAN-Tools\CANreal\Englisch\CANreal_22.en9
Date of print:	2007-06-19

Changes in the chapters

The changes in the document listed below affect changes in the hardware as well as changes in the description of facts only.

Chapter	Changes as compared to previous version
-	Minor revisions.
-	

Technical details are subject to change without further notice.

This page has intentionally been left blank.

Contents

1. Monitor Program CANreal

1.1 Overview

The monitor program CANreal can be run under all supported Windows operating systems (Windows Vista, NT/2000/XP, Windows 95/98/ME)! A similar user interface with a restricted range of functions as for Windows operating systems is available for the supported UNIX operating systems (Linux, LynxOS, PowerMAX OS, SGI-IRIX6.5 and Solaris).

CANreal is a menu-controlled program which is used for monitoring and testing CAN networks. Its self-explaining user interface offers a quick lead-in into the way the CAN network works.

This manual explains in detail the individual menus of CANreal. The description is followed by application examples.

1.2 Program Call

The program is automatically installed with the CAN SDK.
Before calling the program the driver has to be started.

Note: CANreal can run parallel to other CAN-applications and e.g. display the identifiers used there or transmit on any identifier. It is possible to run multiple CANreal instances of the software on the same or different channels.

When the program is called, parameters can be specified in the command line:

Call	Function
`CANreal --start`	CANreal is called and initialised so that the received CAN messages are immediately displayed.
`CANreal` *`profilname`*	CANreal is called and the parameter settings stored under *`profilname`* are adopted.
`CANreal` *`profilname`* `--start`	Combination of both calls above.

Table 1: Calling CANreal via parameters

The creation and meaning of profile files will be described on page 18.

At the change of the previous program name to CANreal the standard profile (formerly called `canscope.ini`) is not transferred. The standard profile has to be renamed to `canreal.ini`.

2. Explanation of the Functions of the User Interface Elements

2.1 Display of the CANreal Window

The CANreal program window is constructed as shown below:

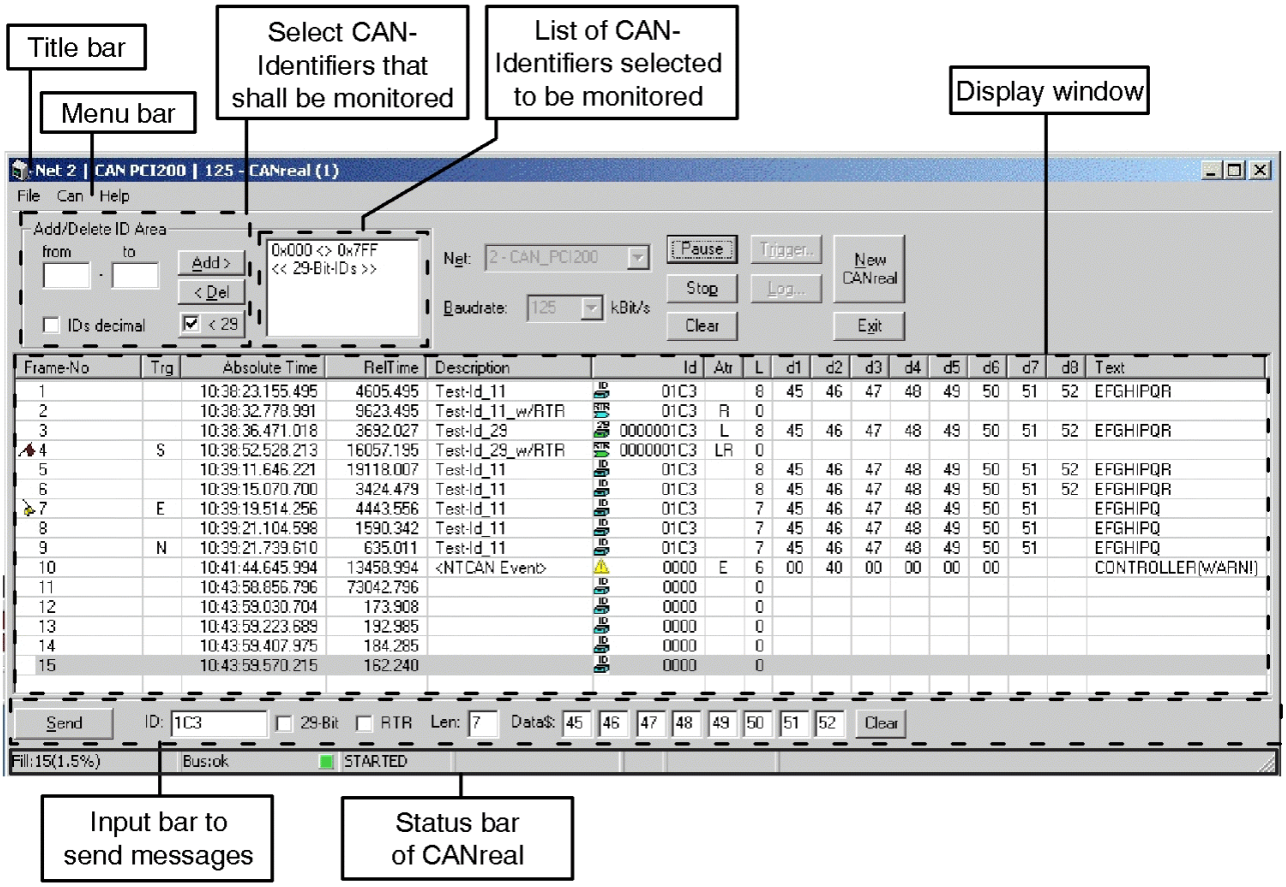


Fig. 1: CANreal program window

2.2 Title bar

In the title bar the following details (from left to right) are displayed:

- Net number (here e.g.: Net 2)
- Module in use (here e.g.: CAN-PCI/200)
- Baud rate (or Auto, as long as the automatic baud rate detection has not been completed) (baud rate used in this example: 125 Kbit/s)
- If Static view is used: STATIC
- If listen-only mode is used: LISTEN ONLY
- Number of the CANreal program instance in brackets (here e.g.: 1)

2.3 Display Windows and Buttons

In the display window the selected CAN messages are displayed.

Column	Format	Meaning
Frm.-No	decimal	message number, each of the messages to be displayed gets a serial number Static mode: number of the frames received with the identifier specified under *Id* and the attribute specified under *Atr*.
Trg	S, E, T, N	trigger condition for CAN-messages (see page 14) S... *Start Trigger* T... *Time Period* E... *End Trigger* N... *Number of Messages*
Absolute Time	HH:MM:SS.msec	time flag (absolute) In this column the time is shown in which the message displayed has been received, counted from the first message received. If HW-Timestamp is active, microseconds are displayed (HH:MM:SS:msec)
RelTime	decimal milliseconds	time flag In this column the time is shown in which the message displayed has been received, counted from the previous message in milli-seconds. The resolution depends on the operating system (> 10 ms). If HW-Timestamp is supported a higher accuracy can be achieved (depending on hardware, ≤ 10 µs). The HW-timestamp is then displayed in milliseconds and microseconds (msec.µsec)
Description	text	description of the identifier If single error diagnostic is active: Display as NTCAN-Event
Id	decimal or hexadecimal	CAN identifier The symbols in this column next to the CAN-Identifier display the attributes of the identifiers: (blue) - 11-bit identifier (green) - 29-bit identifier (blue) - 11-bit Id. with RTR (green) - 29-bit Id., RTR Event frame <NTCANEvent>: 0 controller events 1 baud change (see page 9) 2 ECC-event-ID (only at *single error diagnostic*, see page 22)
Atr	L, R, E	attribute L... 29-bit ID R... RTR bit E... change of status of the CAN bus: *ok, warn* or *error* (for event frames)
L	0...8	Number of valid data bytes of the message.
d1 ... d8	hexadecimal	data bytes, hexadecimal (two digit).
Text	ASCII	The data received are shown in ASCII text

Table 2: Meaning of the entries in the display window

The messages to be displayed are selected via the buttons of the *Add/Delete ID Area* or the respective specification menus. Each message received is listed in a new line (*Scroll Down* button)

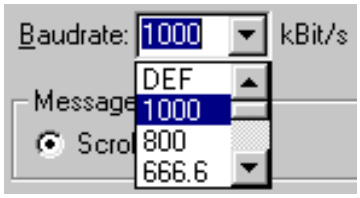
The column width of the table can be changed according to requirements with the left mouse button.

Attention! The time difference is determined by the monitor at the moment in which one or more messages are read out of the driver buffer. The time differences therefore strongly depend on the utilisation of the operating system.

Attention! High accuracy attainable with CAN-boards and drivers with hardware timestamp option.

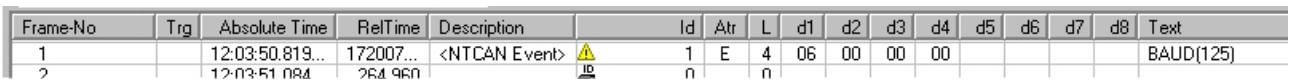
2.4 Description of the Buttons

2.4.1 Baud Rate



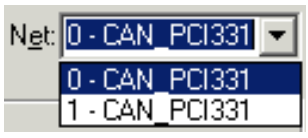
Here you can set the baud rate. If the specification *DEF* is selected, CANreal will operate with the baud rate set via the CAN driver. Alternatively, you can also specify one of the values offered.
If the specification *BRT01* is selected, a four-digit hexadecimal value for the controller register BTR0 and BTR1 can be entered.

An automatic baud rate detection can be selected, if the board supports this function. An overview can be taken from the table in the chapter: “Matrix of Supported Operating Systems and features” of the manual ‘CAN-API, Function Description’.
Select *AUTO* in the field *Baudrate* and start the program via the button *Start*. Now the driver tries to determine a CANopen-standardized baud rate on the CAN bus.
The auto-baud mode will be active, until a standardized baud rate is detected. Then the baud rate is transferred automatically and the program CANreal is run with this baud rate.
A change of the baud rate is always reported as NTCAN event (*Id*: 1, for baud change event). The new baud rate will be shown in brackets in the column *Text*. The current baud rate is displayed in the title bar of the program window.



Frame-No	Trg	Absolute Time	RelTime	Description	Id	Atr	L	d1	d2	d3	d4	d5	d6	d7	d8	Text
1		12:03:50.819...	172007...	<NTCAN Event>	1	E	4	06	00	00	00					BAUD(125)
2		12:03:51.084	264.960		0		0									

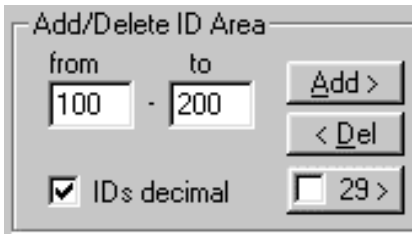
2.4.2 Net



In this input box the network number used for the board can be entered.
The first board in the PC normally has the network numbers 0 and 1 (only the 0 for a one-channel board), every further board has network numbers which are accordingly rising (2 and 3, 4 and 5 ... for one-channel boards 2, 4 ...).
The name of the board is shown with the available network numbers in the list *Net*.

2.4.3 Add/Delete Area

Add/Delete ID Area (11-bit identifier)



Here the lower and upper limit of the ID area to be activated or deactivated is specified (max: 0_h -$7FF_h$). It is permissible to activate or deactivate ID areas which are already active or inactive. The IDs have to be specified with hexadecimal values, if not selected differently (see button *IDs decimal*).

Add >
< Del

By means of these buttons the ID area which is shown in the input boxes is activated or deactivated and shown in the display window.

IDs decimal

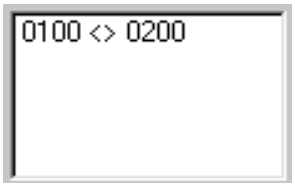
After starting the monitor all IDs are shown hexadecimally. By pressing this button all IDs in the window will be shown decimally and the hexadecimal values will be replaced by decimal values in all ID input boxes.

29-Bit Identifier

If you activate this function, you can also transmit and receive messages with 29-bit CAN identifiers via CANreal. Of course, the hardware and CAN driver have to be able to work with 29-bit CAN identifiers.

If the function has been activated, the menu item *29-Bit* appears in the specification line for transmit messages. In order to transmit 29-bit frames it has to be activated.

Active ID's



In this window the active ID areas are shown. A single click with the left mouse button accepts the value range to the specification fields *from* and *to*. A double click deletes the selected ID area.

2.4.4 Start/Stop-, Trigger-Buttons

Start/Pause

Clicking the *Start*-button starts the listing of received messages in the display window. Clicking the *Start* button again switches to *Pause*. In this status the recording of the received messages pauses without stopping the trigger- and logging function. Clicking the *Start/ Pause*-button again ends the pause and the listing of the received data continues.

Stop

Clicking the *Stop*-button stops the recording and the listing of received messages in the display window and the trigger- and logging-functions are completed.

Clear

Clicking the *Clear*-button deletes all CAN message objects in the window and the message counter is reset to zero.

Trigger

Clicking this icon opens the dialogbox *Trigger and Logging* and conditions can be specified, which start (*Start Trigger*) or end (*End Trigger*) the recording of the messages in files. In addition properties of the file in which the messages are stored can be specified with *Output.*

Log

Clicking the *Log*-button opens the *Output* directly. The properties of the file, in which the messages are stored can be specified here.

Exit

Clicking this button closes the file and ends the program CANreal.

New CANreal

Clicking the *New CANreal* button starts a new instance of the program and opens a further CANreal program window. The number of the instance will be shown in the title bar.

2.5 Input Bar for Send Messages

The data of the CAN-messages to be transmitted have to be specified in the input bar at the lower window frame. The same data formats as for the display window are valid. Values out of the valid value range are not accepted.

Send

By clicking this button the data specified in the input bar are transmitted.

Clear

By clicking this button once the specified transmit data in the input bar for transmit data are deleted. By clicking this button twice the entries for CAN identifier and length are deleted as well.

2.6 Status Display

This status bar shows the status of the program.

Fill: total number of messages in the list (fill level in %)

Bus: Bus status (ok, WARN!, OFF!) shows the status of the CAN controller

Colour of the status field:	Status:
green	ok
yellow	WARN!
red	OFF!

(...-Lost:n) loss of n messages
n... number (see page 27)

Status:
- STARTED — recording is started
- PAUSED — recording pauses
- STOPPED — recording is stopped

TRIG:
- WAITING — waiting for trigger
- TRIGGERED — trigger started
- DELAYED — end trigger occurred, but messages are still recorded
- ENDED — end trigger occurred

LOG: the option *Log to File* is active

In the listen-only mode the background of the status bar is highlighted depending on the colour schema used.

2.7 Dialogbox Trigger and Logging

2.7.1 Start Trigger

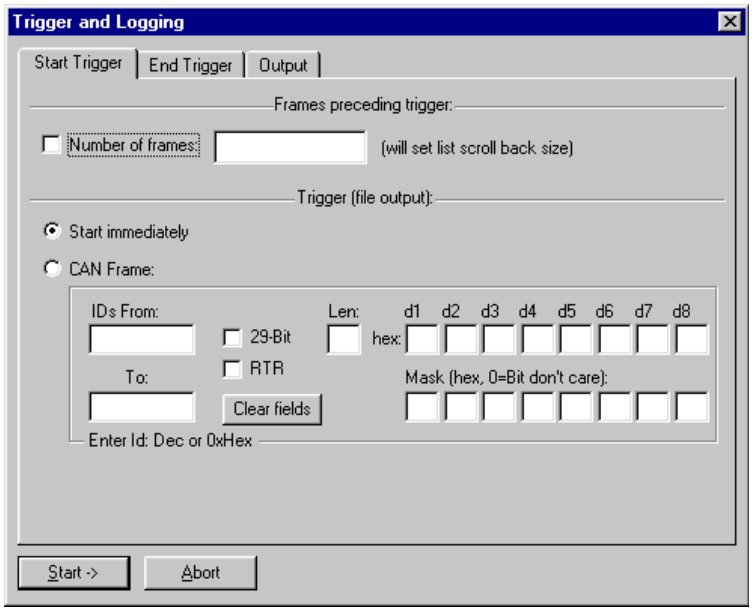


Fig. 2: Start trigger conditions

The conditions which start (trigger) the recording of the messages in a file can be specified here.

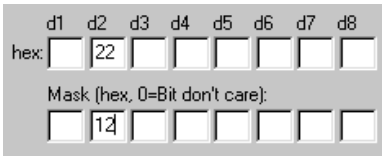
In the field *Frames preceding Trigger* a *Number of Frames* (decimal) that precede the trigger can be specified. The preceding CAN messages are stored to a file: `*-pretrig.csplog`

Start immediately starts the recording of the CAN messages directly after triggering.

CAN Frame starts the recording with CAN messages with specified properties. The selection can be done by identifier, 29-bit identifier, active RTR-bit, the length of the message (*Len*: 0-8) and with defined data bytes (d1-d8). Only the fields with an entry restrict the selection. If RTR is chosen, the entries of the length and the data are always invalid.

In addition the selection with a data-bit mask is possible. Only specified data bytes *d1-d8* must correspond to the bytes specified. The chosen data bits have the value '1' in the mask.

Example for the selection with data-bit mask:



specified data byte $d2 = 22_h$ (0010 0010 $_b$)
data byte mask $M2 = 12_h$ (0001 0010 $_b$)

d2:	0	0	1	**0**	0	0	**1**	0	
M2:	0	0	0	**1**	0	0	**1**	0	
CAN-Data:	x	x	x	0	x	x	1	x	→ Condition fulfilled
	x	x	x	1	x	x	1	x	→ Condition not fulfilled

The data bits of the CAN data selected with '1' in the mask M2 agree with the data bits of d2 only in the upper example. The data bits of the CAN data marked with x are not relevant for the evaluation because of the mask M2.

2.7.2 End Trigger

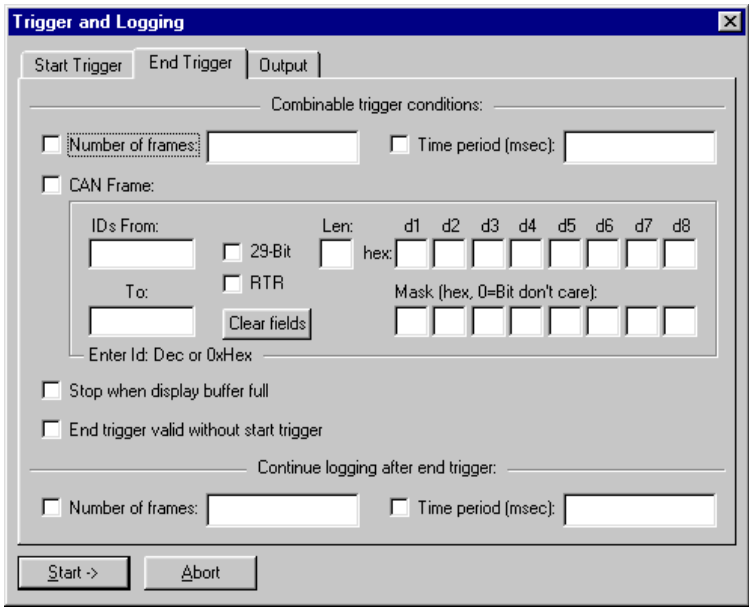


Fig. 3: End trigger conditions

In this window the conditions can be specified which stop the recording of the messages in files.

The recording can be stopped after a specified *Number of frames* or a specified *Time period*.

CAN Frame stops the recording after CAN messages with specified properties.

The selection can be done by identifier, 29-bit identifier, active RTR-Bit, the length of the messages and with defined data bytes (d1-d8). Only the fields with an entry restrict the selection. If RTR is chosen, the entries of the length and the data are always invalid.

In addition the selection with a data-bit mask is possible. Only specified data bytes *d1-d8* must correspond to the bytes specified (see example on page 14).

Stop when display buffer full stops the recording if the display window is full.

With *End trigger valid without start trigger* the end-trigger condition is valid even if the start-trigger condition has not already occurred.

In the field *Continue Logging after end trigger* a *Number of Frames* or a *Time period* can be specified, in which the recording is continued after the trigger condition occurred. But the log file will not be closed after *Time period*, if no CAN frame is received.

2.7.3 Output

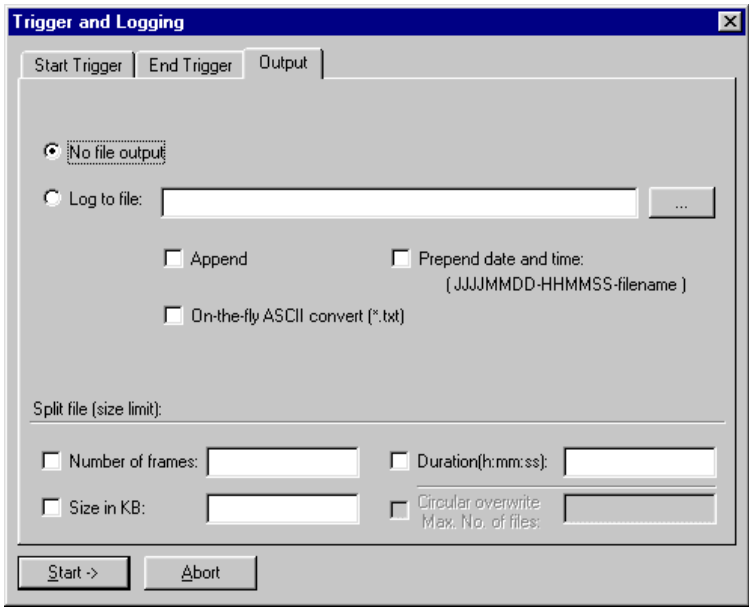


Fig. 4: Properties of the output file

In this window the properties of the message record file can be specified.

If the messages shall not be recorded into a file the button *No file output* must be chosen.

If the recording of the data in a file is selected with *Log to file*, the file name can be entered directly into the input box or selected with the button next to it.
The selected file has always the format CANreal log files (binary data records) with the extension: `*.csplog`.

Append does not overwrite the existing file when the recording is started again but the recording in the file is continued, i.e. a new received CAN message is appended at the end of the file.
The option *Append* can not be selected together with the options *Prepend date and time* and/or *Circular overwrite* and will be hidden if one of these selections is done.

Selecting *Prepend date and time* prepends the time to the file name in the format `JJJJMMDD-HHMMSS` (4 digits for the year, 2 digits each for month, day, hours, minutes and seconds).

Selecting *On-the-fly-ASCII convert (*.txt)* converts the received data directly into text and records it to a text file parallel to binary recording. This process needs a higher computing power.

> **Note:** If the computing capacity is insufficient, messages might get lost and the conversion will be delayed.
>
> It is safer to record the binary messages and to convert the binary file afterwards with *Convert log files to text* into a readable text file (see page 20).

In the *Split file* field the size of the files can be limited. A maximum *Number of Frames* and a maximum *Size in KB* can be selected.
The *Duration* of the file can be specified in hours, minutes and seconds. *Duration* does not affect the pretrigger files (`*-pretrig.csplog`). The pretrigger files can only be limited with the *Number of frames* or the *Size in KB*.

> **Note:** If *Duration* is selected, and even if the time has expired, no new file will be started until a CAN-fame is received.

Because of the file size limit the data are not recorded in a single file but in a corresponding number of size limited files. The file names are composed of a freely selectable name which is consecutively numbered:

```
filename_00000.csplog
filename_00001.csplog
...
filename_nnnnn.csplog
```

The data in the log file header are initialized only once. Except for the file number they are static also for *Split*-files.

The number of the files can be limited as described below:

If at least one option to limit the file size is selected in the field *Split file (size limit)*, the option *Circular overwrite Max. No. of files* can be selected.

With *Circular overwrite* a maximum number of files can be specified.

When the last of the files is written, the recording is continued with overwriting the first and then the following files as e.g. in a ring buffer.

If *Circular overwrite Max. No. of files* is active, *Append* can not be selected. After a new start incoming CAN telegrams can no longer be written to the end of the file.
Circular overwrite does not affect pretrigger files (`*-pretrig.csplog`).
Always the entire file will be stored.

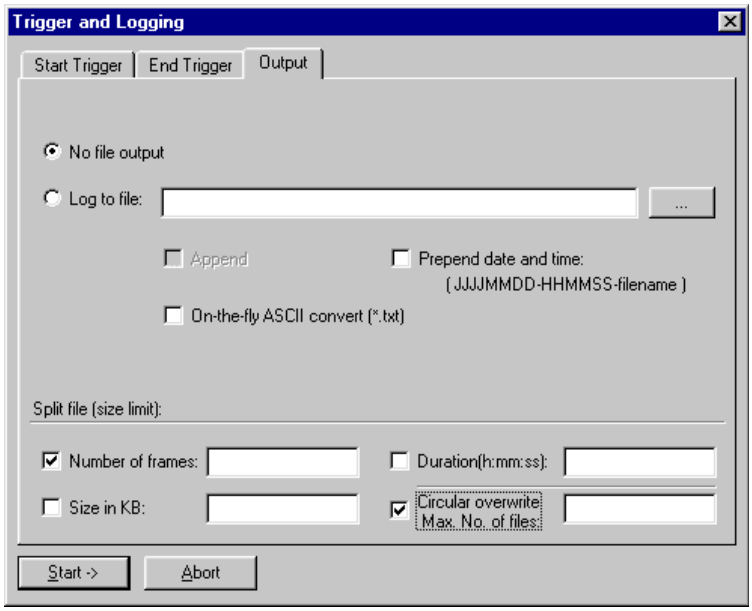


Fig. 5: Size limit of output file

Start->

To start the recording of the messages click the *Start*-button.

Abort

To quit the dialogue box without starting the recording click *Abort*. The actual dialogue-settings will be kept unchanged.

3. Menu

The main menu contains the three menu items *File, CAN* and *Help*:

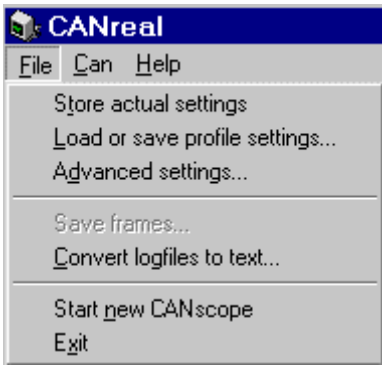


Fig. 6: Menu bar of CANreal and subordinated items of menu *File*

3.1 Menu Item *File*

3.1.1 Store Actual Settings

By calling this menu item the current settings are stored as default settings. When CANreal is called again, all settings correspond to those which were specified at the moment *Store Actual Settings* was called.

3.1.2 Load or Save CANreal Profile

The CANreal parameters specified can be stored in profile files and can be loaded again at a later time.

The profile files have the extension `*.cspini`

With the *start*-check box in the *Action*-field you can select an automatic start after loading.

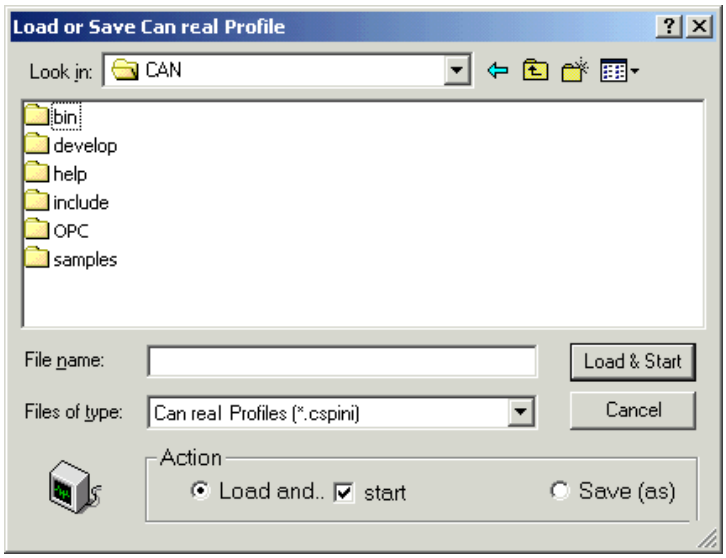


Fig. 7: Load or save CANreal Profile

3.1.3 Advanced Settings/Application

In this menu item in the field *Text description mapping for CAN identifiers* a description file can be assigned to every identifier. The file name has the extension `*.txt` and can be written in the input field, selected with the button next to the input field or generated with *Generate pattern file*.

With *Memory allocation for scroll back and trigger* the *Maximum Number of CAN-frames* (1.000 - 2.000.000 messages) in the scroll-down list can be specified.
If the maximum number is reached, i.e. if the list is full, the first entry of the list will be deleted and the following entries move up.
Now a further entry can be written at the end of the list.

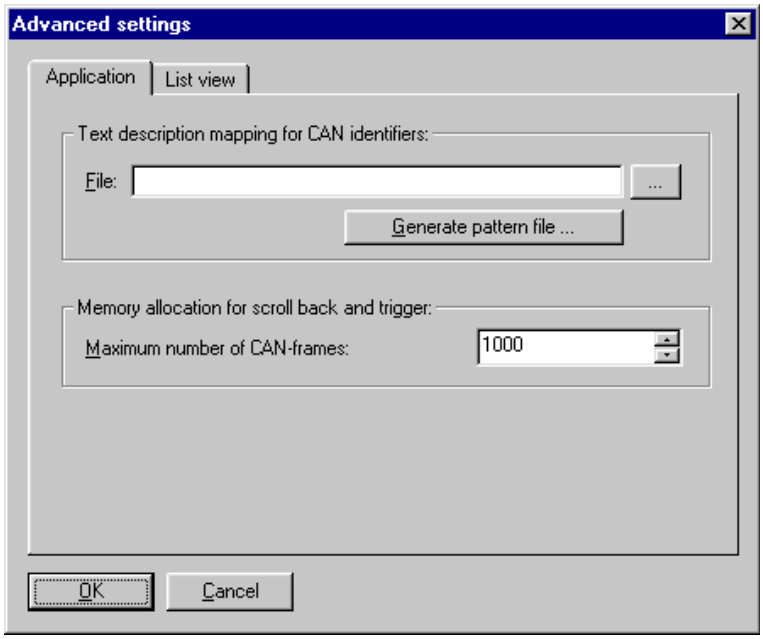


Fig. 8: Advanced settings / Application

3.1.4 Advanced Settings/List View

In this menu item the display of the scroll-down list can be changed.

In the box *Columns visible* single columns of the list can be selected or deselected.
The button *Show all* selects and shows all columns listed. The button *Hide all* deselects and hides the columns listed.
CAN-Id&Data are always displayed.

The columns can be moved with the mouse in the table. They can be arranged in the table as required.

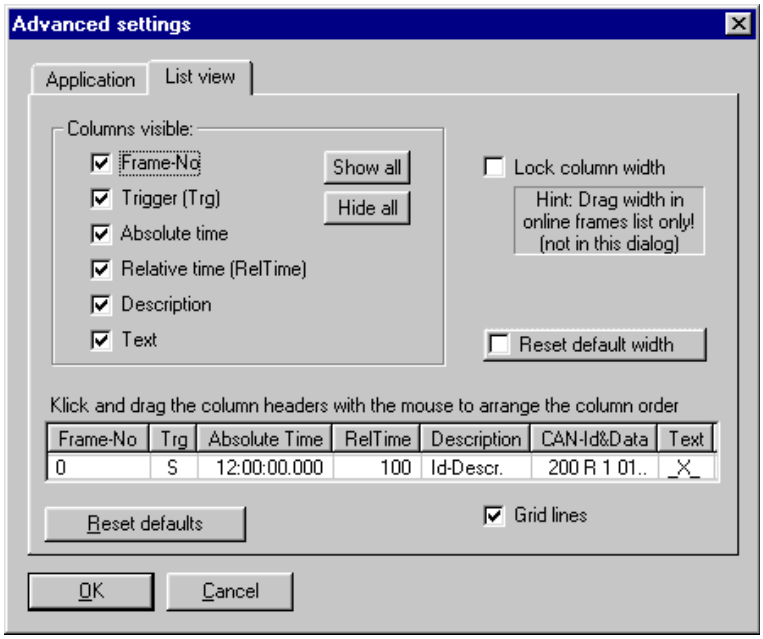


Fig. 9: Advanced settings / List view

3.1.5 Save Frames

Save Frames stores all messages shown in the display window in a binary log file (`*.csplog`) and a readable text file (`*.txt`).

The file type log files (`*.csplog`) has to be selected for saving and the text file will be generated automatically.

If in addition messages are to be stored which are usually deleted, if the window spills over, they have to be stored via *Start File Logging*.

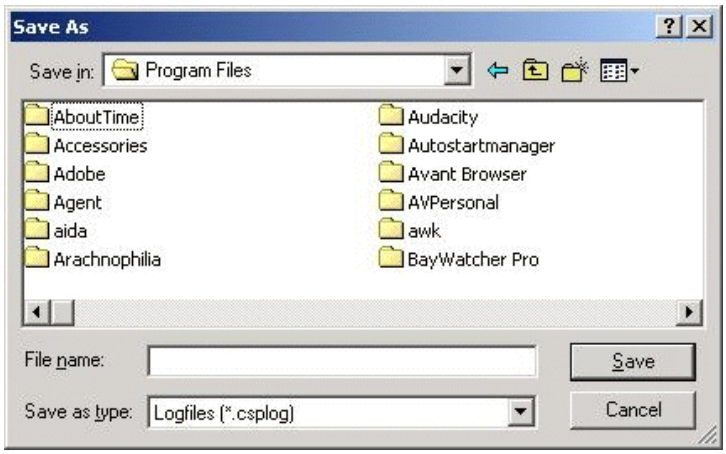


Fig. 10: Save frames as ...

3.1.6 Convert Log Files to Text

With *Convert log files to text* an existing log file (`*.csplog`), with binary coded stored messages, can be selected.
Received data are stored binary coded. The selected log file can be converted to a readable text file (`*.txt`) .

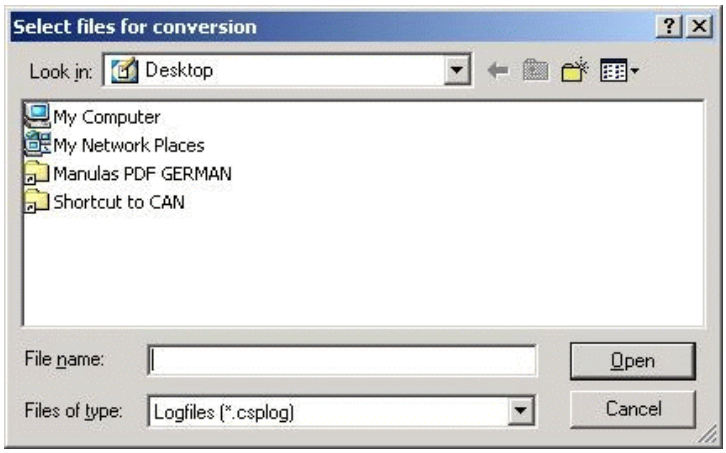


Fig. 11:
Selection of Files

3.1.7 Start new CANreal

Start new CANreal starts a new instance of the program and a new CANreal window will be opened.

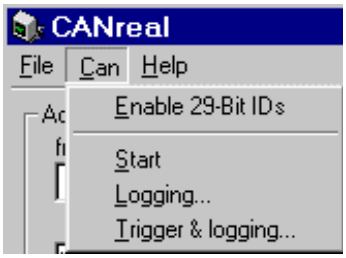
3.1.8 Start COBview

The program COBview can be called via the menu item *Start COBview*.

3.1.9 Exit

Exit the program CANreal.

3.2 Menu item *CAN*



The commands listed below can be activated directly via the corresponding push button or via the menu item CAN:

- **Enable 29-Bit IDs**
- **Start**
- **Logging**
- **Trigger&logging**

See also chapter: 'Description of the Buttons' continued from page 9.

3.2.1 Static View

The static mode can be activated with this menu item. In this mode the frames are not listed in the display window according to the date of their reception, but for every received ID a row is created in which all received frames with this ID are counted. RTR, event frames and 29-bit IDs are counted separately. Switching into the static mode is only possible if the program run is stopped. It is not possible while the program is running.

Frame-No	Trg	Absolute Time	RelTime	Description	Id	Atr	L	d1	d2	d3	d4	d5	d6	d7	d8	Text
3		10:48:35.027.882	8437.107	Test-Id_11	451		7	45	46	47	48	49	50	51		EFGHIPQ
1		10:47:56.966.605	0.000	Test-Id_11_w/RTR	451	R	0									
1		10:48:00.388.418	0.000	Test-Id_29	451	L	8	45	46	47	48	49	50	51	52	EFGHIPQR
1	S	10:48:09.050.325	0.000	Test-Id_29_w/RTR	451	LR	0									
1		10:47:44.110.230	0.000	<NTCAN Event>	0	E	6	00	40	00	00	00	00			CONTROLLER(WARN!)

Fig. 12: Part of the display window in the static mode

Frame-No (in static mode) number of received frames with the identifier *Id* and the attributes listed in the column *Atr*.

3.2.2 Listen-Only Mode (*Controller listen only)*

The listen-only mode can be activated with the menu item *Controller listen only*. This applies only to boards which support the listen-only mode. An overview of this can be taken from the table in the chapter: "Matrix of Supported Operating Systems and Features" in the manual 'CAN-API, Function Description'.

Listen-only mode:

- In this mode frames can only be received, but not transmitted.
- The CAN controller is switched to passive. It does not transmit any acknowledges or error-frames, i.e. the CAN board acts as if it is inexistent.
- There must be at least **two** further participants on the CANbus.

In the listen-only mode the background of the status bar is highlighted (as the tooltip) depending on the colour schema used.

3.2.3 HW-Timestamp

With the menu item *HW-Timestamp* the hardware timestamping can be activated for current boards and current drivers (for further information see chapter: “Matrix of Supported Operating Systems and Features” in the manual ‘CAN-API, Function Description’)

If the hardware timestamp is not activated, the times are specified under *Absolute Time* and *RelTime* with millisecond accuracy (depending on the operating system, actual accuracy ≥ 10 ms). If the timestamping is activated, the times are specified with microsecond accuracy (depending on hardware, actual accuracy ≤ 10 µs). *(Absolute Time:* HH:MM:SS.ms.µs, *RelTime*: ms.µs).
The *RelTime* indicates the HW-timestamp.

Please note that in the HW-timestamp mode at long runtimes the absolute time can differ from the PC-time.

3.2.4 *Single error diagnostic*

The menu item *Single error diagnostic* can only be selected for boards that support this function (for further information see chapter: “Matrix of Supported Operating Systems and Features” of the manual ‘CAN-API, Function Description’).
If the single error diagnostic is active, errors of the single CAN-frames are indicated as event messages.

Frame-No	Trg	Absolute Time	RelTime	Description	Id	Atr	L	d1	d2	d3	d4	d5	d6	d7	d8	Text
7474		12:12:10.178...	1.225		1157		8	01	02	03	04	05	06	07	08	_____
7475		12:12:10.179...	1.176		848		8	01	02	03	04	05	06	07	08	_____
7476		12:12:10.179...	0.368	<NTCAN Event>	2	E	4	1C	A6	56	05					ECC:"stuff error"
7477		12:12:10.179...	0.053	<NTCAN Event>	2	E	4	1C	F3	5E	05					ECC:"other type of error"
7478		12:12:10.180...	0.153	<NTCAN Event>	2	E	4	1C	A2	5F	05					ECC:"stuff error"

Fig. 13: Part of the display window with listing of single errors

In the column *Description* the frame is described as NTCAN Event. In the *Text* column the single errors read for the specific CAN frames from the ECC-error register of the CAN controller are described.

The column *Id* indicates the warning symbol for the error and the event-Id: 2, which encodes the ECC-Events.
The second byte (*d2*) contains the number of the errors in the ECC-register, which is defined in the manual of the according CAN-controller.

3.3 Menu item *Help*

The pull-down menu of the menu item *Help* of the main menu contains the menu items *About* and *Help browser*, as described below.

3.3.1 CANreal-Information Window *About*

With the menu item *About* in the pull-down menu of the menu item *Help* the *About CANreal*-information window is opened.

The window shows program and hardware information for the respective CAN net (here: Net 2) as the CANreal-version, the hardware and firmware and the driver and NTCAN-versions.

Features gives the supported features coded as hexadecimal number.
The abbreviations in brackets stand for:

2b	CAN 2.0B-support
Ts	Timestamping is supported
Sd	Smart Disconnect is supported
Lo	Listen-only mode is supported

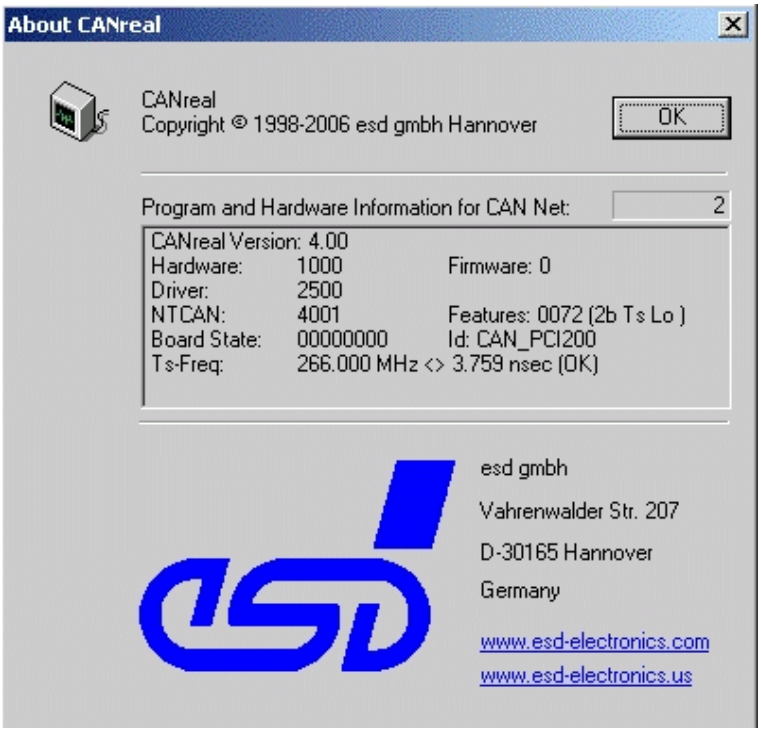


Fig. 14: Program and hardware information

Further information about the functions can be taken from the manual 'CAN-API, Function Description'.

The *Ts-Freq:* is the timestamp frequency, if the board and the driver support the timestamping (see page 22). The timestamp-frequency is given in MHz and the resolution in nanoseconds.
The following *(OK)* indicates that board and driver both support the timestamping.

3.3.2 *Help browser*

Selecting the menu item *Help-Browser* in the pull-down menu of the menu item *Help* opens the help-browser of CANreal. The help-browser can also be opened with the control key F1.

4. Examples

4.1 Example of a Log File

```
[--] ---------------------------------------------
[--] CANreal Log File Header (created)
[--] ---------------------------------------------
[F1] Source:                C:\TEMP\CANreal logtest\log.csplog
[F2] Split file series:     log.csplog
[F3] File serial number:    -1
[F4] Id-Description file:   C:\TEMP\CANreal Logtest\Id2Description.txt
[F5] Header type:           0
[F6] Header length:         1835
[F7] File date:             27.06.2003
[F8] File time:             09:45:37
[S1] Hardware:              1100
[S2] Driver:                2400
[S3] Firmware:              0C02
[S5] dll:                   3100
[S6] Board:                 CAN_PCI331
[S7] status:                00000000
[S8] OS:                    Windows 2000 - Service Pack 3 (5,0,2195,2,[Service Pack 3])
[S9] CANreal:               2.30
[SA] Features:              0072 ( 2b Ts Lo )
[SB] HW Time stamps:        resolution 266.000 MHz <> 3.759 nsec (OK)
[C1] Net number:            0
[C2] Baud:                  500
[C3] 29-bit-Ids:            0 - disabled
[T0] Start trigger:         frames_preceding(on,1) can_frame(on) id_from(513) id_to(-1) len(-1)
                29-Bit(off) rtr(off) data(--, --, --, --, --, --, --, --)
                mask(--, --, --, --, --, --, --, --)
[T1] End   trigger:         number_frames(off,2) time_period(off,3000) stop_full(off)
                end_w/o_start(off) post_frames(on,4) post_time(off,5000) can_frame(on)
                id_from(513) id_to(-1) len(-1) 29-Bit(off) rtr(on)
                data(--, --, --, --, --, --, --, --)
                mask(--, --, --, --, --, --, --, --)
[C4] Id-Area:               0000-2047 ($000-$7FF)
[--] ---------------------------------------------

Frame_Number  Trigger  Absolute_Time  Msec  Relative_Time  Description  CAN_Id_dec  CAN_Id_hex
Attributes  Length  d1  d2  d3  d4  d5  d6  d7  d8  Text

Num Trig  Abs     Msec Rel  Desc          IdDec  IdHex Attr Len d1  d2  d3  d4  d5  d6  d7  d8   Text
17  S    09:45:51 423 851   Trig. Start F  513   0201       8   00  00  00  00  00  00  00  00    __
18       09:45:52 204 781   Range Control  709   02C5       1   01                                _
19       09:45:52 214 10    Pressure       710   02C6       1   01                                _
20       09:45:53 847 1633  Range Control  709   02C5       1   01                                _
21       09:45:53 857 10    Pressure       710   02C6       1   01                                _
22       09:45:55 489 1632  Range Control  709   02C5       1   01                                _
23       09:45:55 509 20    Pressure       710   02C6       1   01                                _
24       09:45:57 141 1632  Range Control  709   02C5       1   01                                _
25       09:45:57 152 10    Pressure       710   02C6       1   01                                _
26  E    09:45:58 283 1132  Trig. Stop F.  513   0201  R    0
27       09:45:58 784 501   Range Control  709   02C5       1   01                                _
28       09:45:58 794 10    Pressure       710   02C6       1   01                                _
29       09:46:00 426 1632  Range Control  709   02C5       1   01                                _
30  N    09:46:00 436 10    Pressure       710   02C6       1   01                                _
```

Trigger Start Frame starts the recording of the messages. *Trigger Stop Frame* marks the end of the recording. Because as *end trigger* condition with *Continue logging after end trigger* a number of messages following *Trigger Stop Frame* is selected, four further messages are recorded.

4.2 Example of a Log File with *Frames preceding Trigger*

In this example *Frames preceding Trigger* has been selected in *Start Trigger* for a log file. Therefore the messages preceding the message which fulfills the trigger conditions have been recorded in this `*-pretrig.csplog`-file.

```
[--] ---------------------------------------------
[--] CANreal Log File Header (created)
[--] ---------------------------------------------
[F1] Source:                      C:\TEMP\CANreal logtest\log-pretrig.csplog
[F2] Split file series:           log.csplog
[F3] File serial number:          -1
[F4] Id-Description file:         C:\TEMP\CANreal Logtest\Id2Description.txt
[F5] Header type:                 0
[F6] Header length:               1835
[F7] File date:                   27.06.2003
[F8] File time:                   09:45:37
[S1] Hardware:                    1100
[S2] Driver:                      2400
[S3] Firmware:                    0C02
[S5] dll:                         3100
[S6] Board:                       CAN_PCI331
[S7] status:                      00000000
[S8] OS:                          Windows 2000 - Service Pack 3 (5,0,2195,2,[Service Pack 3])
[S9] CANreal:                     2.30
[SA] Features:                    0072 ( 2b Ts Lo )
[SB] HW Time stamps:              resolution 266.000 MHz <> 3.759 nsec (OK)
[C1] Net number:                  0
[C2] Baud:                        500
[C3] 29-bit-Ids:                  0 - disabled
[T0] Start trigger:               frames_preceding(on,1) can_frame(on) id_from(513) id_to(-1)
                                  len(-1) 29-Bit(off) rtr(off)
                                  data(--, --, --, --, --, --, --, --)
                                  mask(--, --, --, --, --, --, --, --)
[T1] End   trigger:               number_frames(off,2) time_period(off,3000) stop_full(off)
                                  end_w/o_start(off) post_frames(on,4) post_time(off,5000)
                                  can_frame(on) id_from(513) id_to(-1) len(-1) 29-Bit(off)
                                  rtr(on)data(--, --, --, --, --, --, --, --)
                                  mask(--, --, --, --, --, --, --, --)
[C4] Id-Area:                     0000-2047 ($000-$7FF)
[--] ---------------------------------------------
Frame_Number  Trigger  Absolute_Time  Msec  Relative_Time  Description  CAN_Id_dec  CAN_Id_hex
Attributes  Length  d1  d2  d3  d4  d5  d6  d7  d8  Text
Num Trig  Abs      Msec  Rel    Desc          IdDec  IdHex  Attr  Len d1  d2  d3  d4  d5  d6  d7  d8  Te
                                                                                                    xt
1       09:45:39  055   11496  Range Control  709   02C5        1   01                              _
2       09:45:39  055   0      Pressure       710   02C6        1   01                              _
3       09:45:40  698   1643   Range Control  709   02C5        1   01                              _
4       09:45:40  698   0      Pressure       710   02C6        1   01                              _
5       09:45:42  340   1642   Range Control  709   02C5        1   01                              _
6       09:45:42  350   10     Pressure       710   02C6        1   01                              _
7       09:45:43  983   1632   Range Control  709   02C5        1   01                              _
8       09:45:43  993   10     Pressure       710   02C6        1   01                              _
9       09:45:45  625   1633   Range Control  709   02C5        1   01                              _
10      09:45:45  635   10     Pressure       710   02C6        1   01                              _
11      09:45:47  277   1642   Range Control  709   02C5        1   01                              _
12      09:45:47  277   0      Pressure       710   02C6        1   01                              _
13      09:45:48  920   1642   Range Control  709   02C5        1   01                              _
14      09:45:48  930   10     Pressure       710   02C6        1   01                              _
15      09:45:50  562   1633   Range Control  709   02C5        1   01                              _
16      09:45:50  572   10     Pressure       710   02C6        1   01                              _
17  S   09:45:51  423   851    Trig. Start F  513   0201        8   00  00  00  00  00  00  00  00  __
```

4.3 Description of the Header Lines [F1]...[C4] Considering as an Example

CANreal Log File Header (appended)		The header and the following CAN messages were appended to the log file.
[F1] Source:	C:\TEMP\CANreal logtest\log.csplog	Name of the binary log file which has been converted.
[F2] Split file series:	log.csplog	Base name of the file or the split file series without number- or pretrig-affix. If not a split file series, the same as name of log file.
[F3] File serial number:	-1	Serial number of a split file series. If no split file series =-1
[F4] Id-Description file:	C:\TEMP\CANreal logtest\Id2Description.txt	File which contains the description of the CAN identifiers selected for conversion.
[F5] Header type:	0	(program-internal data record type number)
[F6] Header length:	1835	(program-internal data record length)
[F7] File date:	29.07.2003	Date of binary-log file which is converted.
[F8] File time:	12:25:27	Time of binary-log file which is converted.
[S1] Hardware:	1100	Hardware version of the CAN board (information of the CAN driver).
[S2] Driver:	2400	CAN-driver version (information of the CAN driver)
[S3] Firmware:	0C02	Firmware version of the CAN board (information of the CAN driver).
[S5] dll:	3100	Version of NTCAN.DLL (information of CAN driver).
[S6] Board:	CAN_PCI331	Board version of the CAN board (information of the CAN driver).
[S7] status:	00000000	Status of the CAN board (information of the CAN driver).
[S8] OS:	Windows XXXX - Service Pack X (xxxxx,[Service Pack X])	Windows operating system version (information of operating system).
[S9] CANreal:	X.XX	CANreal program version
[SA] Features:	0072 (2b Ts Lo)	As described under *Help / About* (see page 23)
[SB] HW-Timestamp:	resolution. ... (OK)	HW-timestamp supported and selected
[C1] Net number:	0	CAN net number
[C2] Baud:	500	CAN baud rate
[C3] 29-bit-Ids:	1 - enabled	CAN option: 29-bit-identifier active.
[T0] Start trigger:	none	Dialogue options for start trigger as text.
[T1] End trigger:	none	Dialogue options for end trigger as text.
[C4] Id-Area:	0000-1000 ($000-$7FF)	...
[C4] Id-Area:	2000-2047 ($000-$7FF)	Active CAN Id-ranges

4.4 Example of a Log File with *Lost frames*

Lost-frames result, if CAN messages are received faster than CANreal can read them out of the FIFO buffer of the driver. The CAN driver discards some messages. They are marked with "lost" behind the message number.

The number of the *lost-frames* lies in the range between 1 and 255 messages. 255 means 255 or more messages. After a ‘lost’ message the counting starts again with 0.

Exemplified extract of a log file with *lost-frames*:

590585	11:57:41	143	=0	Frame_1_1920_CanId	1920	0780	3	71	04	41	q_A
590586 - 89 lost	11:57:41	143	=0	Frame_1_1920_CanId	1920	0780	3	71	04	9B	q_›
590587	11:57:41	153	10	Frame_1_1920_CanId	1920	0780	3	71	04	9C	q_œ
...											
616053	11:57:43	176	=0	Frame_1_1920_CanId	1920	0780	3	71	04	16	q__
616054 - 255 lost	11:57:43	176	=0	Frame_1_1920_CanId	1920	0780	3	71	04	F9	q_ù
616055	11:57:43	186	10	Frame_1_1920_CanId	1920	0780	3	71	04	FA	q_ú
...											
618125	11:57:43	347	=0	Frame_1_1920_CanId	1920	0780	3	71	04	10	q__
618126 - 95 lost	11:57:43	347	=0	Frame_1_1920_CanId	1920	0780	3	71	04	70	q_p
618127	11:57:43	357	10	Frame_1_1920_CanId	1920	0780	3	71	04	71	q_q

To avoid *lost-frames*:

- Close other applications.
- Bring CANreal window in the foreground.
- Increase the application priority with Windows Task Manager (only for experienced user!)
- Insertion of a CAN board to relieve the host-CPU.
- Usage of a fast computer, from 1.5 GHz, normally can manage 100% bus load with 0-byte CAN messages at 1000 Kbit/s. (This is the worst case which usually does not occur.)

4.5 Procedure to Request or Receive Messages

Target of this example is to receive data on the identifiers 0x0E (14 dec) and 0x0F (15 dec). The data are to be requested by transmitting an RTR message.

The example can only work, if a CAN participant with the correct baud rate has been connected and the identifier used is able to respond with remote requests!

Procedure:

1. If necessary, change the values for bit rate and network number and click the button *Del>* to delete the entries in the list of CAN identifiers that shall be displayed.

2. Do not (!) select *IDs decimal* !

3. Delete the entries in the list of CAN identifiers with clicking the *Del<* button. Enter ‘E’ and ‘F’ (without leading 0x) into the fields *Low Limit* and *High Limit*. Click the *Add>* button

4. Start.
   Received messages on 0x0E or 0x0F are now displayed.

5. Enter ‘E’ (without leading 0x) in the *ID*- field in the specification line for transmit messages and click on the *RTR*- field.

6. After pressing the *Send* button the CAN messages can be seen in the message window.

7. If now the decimal display is activated, all relevant values in the input boxes and display windows will be renewed and the IDs of the messages received from that time on will be shown decimally.